Représentation orientée navigation d'environnements à grande échelle

Patrick Rives, Romain Drouilly, Tawsif Gokhool

Représentation orientée navigation d'environnements à grande échelle

P. Rives[1] , R. Drouilly[1,2], T. Gokhool[1]

[1] INRIA
2004 route des Lucioles BP93
09902 Sophia Antipolis cedex
Prénom.Nom@inria.fr

[2] ECA Robotics
ZI Toulon Est
262 Rue des Frères Lumière
83130 La Garde

Résumé

Nous présentons une nouvelle représentation hybride métrique/topologique/sémantique appelé MTS-Maps conçue pour la conduite automatique de véhicule autonome dans des environnements de grande taille d'intérieur ou d'extérieur. Basée sur une représentation générique, la sphère RGB-DL, elle permet de s'affranchir de la technologie du capteur utilisé. Ces différentes couches permettent de traiter le problème de localisation allant de la requête contextuelle jusqu'au calcul de l'erreur de pose utilisée dans le feedback du contrôleur. Les résultats présentés montrent ses performances à la fois au niveau métrique par une expérimentation de conduite automatique dans un environnement urbain et au niveau sémantique par une comparaison avec une implémentation récente d'une méthode Bag-of-Words.

Mots Clef

approche directe pour le SLAM visuel, cartographie 3D dense, Coarse-to-fine localisation, indexation d'images

Abstract

A new hybrid metric-topological-semantic map structure, called MTS-map, is presented that allows a fine metric-based navigation and fast coarse query-based localisation. Based on a generic representation, the RGB-DL spherical view, it allows to not depend on the sensor technology. The different layers allow us to fully handle the problem of localisation in large scale environments. Results are presented which highlighted the efficiency of the representation both at the metric level for control purposes and at the semantic level by comparing with a standard implementation of a Bag-of-Words method.

Keywords

Direct method for visual SLAM, Dense 3D mapping, Coarse-to-fine localisation, Image retrieval

1 Introduction

La présence de véhicules autonomes partageant notre environnement quotidien est en train de devenir une réalité. Initié par Google avec la Google Car, tous les constructeurs automobiles ont maintenant des projets de voitures dotées de fonction de conduite automatique. Coté aérien, la maitrise de la technologie des petits drônes permet d'envisager leur déploiement à court terme pour des missions de surveillance dans des environnements de type urbains ou à l'intérieur de bâtiments comme des gares ou des parkings. Parallèlement, l'apparition de nouveaux capteurs grand public comme les caméras RGB-D (Kinect, Asus) et l'évolution en terme de puissance de traitement embarqué et de communication des téléphones portables et des tablettes permet d'envisager des implémentations efficaces à bas coût. Naviguer de façon sûre dans des environnements de grande taille, variables au sens de leur contenu ou des conditions d'observation, nécessite d'en construire des représentations adaptées. Idéalement, une représentation basée navigation doit contenir les différentes couches métrique, topologique et sémantique (fig.2) nécessaires à la planification efficace et à l'exécution sûre des déplacements [17]. Dans la pratique, une carte uniquement métrique peut s'avérer suffisante pour réaliser des déplacements dans un environnement de petite dimension et parfaitement contraint. Dans un contexte d'environnement réel, évolutif et de grande taille, les couches topologiques et sémantiques permettent de prendre en compte robustesse et efficacité dans les algorithmes. La couche topologique permet d'enrichir la représentation par l'adjonction d'une structure de graphe contenant des informations d'accessibilité [19]. Elle fournit un premier degré d'abstraction permettant la navigation dans des environnements à grande échelle. Enfin, l'ajout d'information sémantique sur les noeuds du graphe topologique permet d'enrichir la représentation par du contexte indépendant du contenu géométrique et photométrique de la scène la rendant ainsi plus robuste vis à vis des conditions d'observation. Il est important également que ces représentations demeurent valides

au cours du temps et permettent une localisation précise en temps réel.

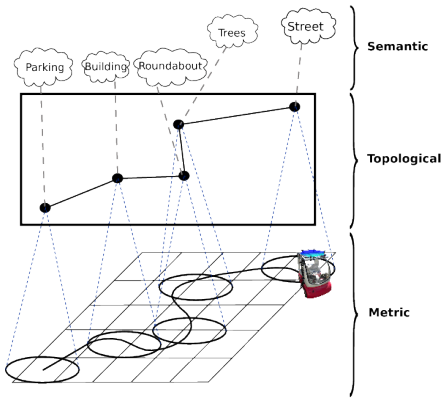


FIGURE 1 – Navigation-based representation

2 Représentation orientée navigation

Se déplacer de façon autonome nécessite de résoudre deux types de problèmes : tout d'abord le calcul d'un itinéraire admissible entre le point de départ et la destination désirée, puis le guidage du véhicule ou de la personne le long de cet itinéraire. Alors que les tâches de planification d'itinéraire ou de trajectoire s'appuient sur une représentation globale reflétant les contraintes d'accessibilité des différents lieux de l'environnement, les tâches de guidage requièrent, elles, une localisation précise relative à l'environnement local. Une représentation orientée navigation devra intégrer ces deux aspects et fournir les mécanismes efficaces permettant de mettre à jour les données tout en conservant la consistance de la représentation. Une première façon d'obtenir une représentation 3D consiste à construire un modèle géométrique de la scène, souvent à base de plans, en utilisant des techniques issues de la communauté graphique. Ce modèle 3D géométrique est rendu photoréaliste en plaquant des textures issues soit de données réelles soit de données synthétiques. Ce type d'approche, largement utilisée dans la conception des jeux vidéos, a été également appliquée à la modélisation d'environnements urbains [8, 3, 13]. En général, ces modèles représentent d'une manière approximative l'environnement et comporte des erreurs de modélisation et des inconsistances photométriques ce qui n'est pas gênant dans une exploitation de rendu visuel mais les rends peu adapté à une exploitation dans un contexte de navigation autonome. De plus, dans le cas d'environnements réels à grande échelle, ils conduisent à des bases de données lourdes dont l'exploitation et la mise à jour sont couteuses en ressources informatiques. Une approche alternative à la reconstruction d'un modèle 3D global, consiste à représenter l'environnement de manière égo-centrée : les approches basées mémoire image proposent de conserver dans la base de donnée directement les données acquises lors d'une phase d'apprentissage, géolocalisées en 2 ou en 3 dimensions dans l'espace. Contrairement aux méthodes globales, ces approches fournissent localement un maximum de précision. En effet, les données sont exprimées dans le repère d'acquisition, ce qui évite de propager les erreurs liées à la reconstruction et aux approximations géométriques des modèles, ce qui améliore la précision de la localisation. Dans [18] une base données d'images clés, contenants des points de Harris ainsi que leur position 3D, est construite lors d'une phase d'apprentissage. [2] proposent une méthode basée sur un graphe d'images générique (image fisheye, omnidirectionnelle) définissant un chemin visuel à suivre. [20] utilise une mémoire image positionnée avec un système GPS, pour de la reconnaissance de lieux et une localisation visuelle basée points d'intérêts.

Nous proposons une nouvelle représentation appelée *MTS-Maps* (Metric-Topologic-Semantic Maps) (fig.2) qui contient à la fois l'apparence (photométrie + géométrie + label) de la scène observée localement et la structure globale de l'environnement. Les différents niveaux de cette représentation seront exploités par les algorithmes de navigation, de la planification du déplacement jusqu'à son exécution par une loi de commande en boucle fermée exploitant en temps réel les images fournies par une caméra embarquée.

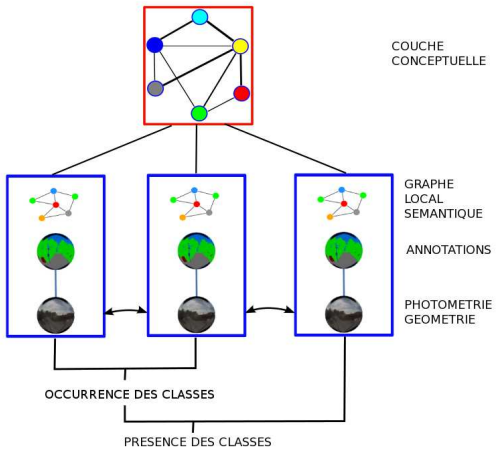


FIGURE 2 – MTS-Map : Les rectangles bleus correspondent aux cartes locales. Elles sont constituées de trois couches correspondant respectivement aux données RGB-D photométriques et géométriques, aux labels associés à l'occurrence des différentes classes et enfin, aux relations entre les différentes régions sous forme d'un graphe sémantique. Le rectangle rouge représentent la couche conceptuelle qui caractérise la connaissance a priori existant entre classes (ex. une voiture est située sur une route). Toutes les cartes locales sont indexées par la présence des classes et leur occurrence.

2.1 Construction des cartes locales

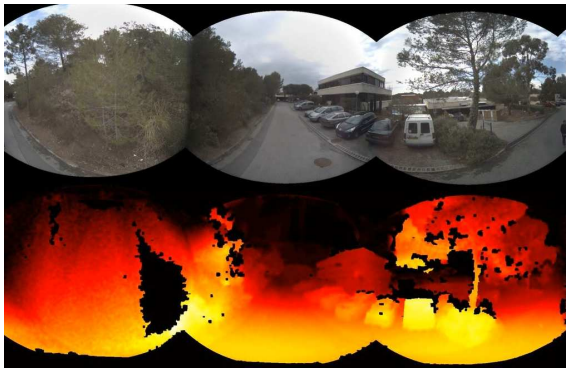
Afin de ne pas être tributaire d'un type de technologie, nous avons choisi de construire notre approche autour d'un modèle de représentation des données perpétuelles indépendant du capteur utilisé qui pourra être différent selon que l'on traite d'application d'intérieur ou d'extérieur. Ce modèle est la sphère RGB-DL, RGB-D représentant la photométrie et la géométrie de la scène et L la labélisation de son contenu. Ce modèle dit égo-centré représente la perception sur 360 degrés de l'environnement local pour une position donnée (x, y, z) de l'observateur.

Image sphérique RGB-D : Deux types de dispositifs ont été développés pour l'acquisition d'images sphériques RGB-D haute résolution en temps réel (30hz). Le premier, dédié aux environnements d'extérieur, utilise un système multi-caméras (fig.3-a) fournissant des images sphériques de résolution 2048X665 (pour plus de détails sur la génération des images sphériques, voir [14]). A chaque pixel est associé une valeur photométrique et la profondeur qui est estimée directement à partir des deux images sphériques au moyen d'un algorithme de mise en correspondance dense [9]. Un deuxième dispositif adapté aux environnements d'intérieur de résolution 3840X640 a été développé à partir de capteur RGB-D Asus Xtion Pro Live basé sur de la lumière structurée (fig.3-b). Un exemple de panoramiques associés aux images sphériques RGB-D fournies par ces capteurs est présenté à la figure 4.

(a) Extérieur (b) Intérieur

FIGURE 3 – Système d'acquisition

Labélisation sémantique : La vue sphérique RGB-D est suffisante pour représenter de façon unique, un environnement statique particulier observé d'une position et dans des conditions d'observation données. Cependant, l'information photométrique est peu robuste aux variations des conditions d'observation et aux perturbations engendrées par la présence d'éléments dynamiques dans la scène. Il est possible



(a) Extérieur

(b) Intérieur

FIGURE 4 – Panoramique RGB-D

d'accroitre cette robustesse en rajoutant une couche de représentation contextuelle par le biais d'informations sémantiques. Cela est fait grace à un algorithme de classification qui associe à chaque pixel un label d'appartenance à une classe préalablement définie. Cette classification se fait en deux étapes. Tout d'abord des descripteurs locaux (SIFT + couleur) sont extraits de façon dense dans l'image et la distribution de probabilité de chaque classe est estimée au moyen d'une méthode de *Random Forest* [5]. Bien que performante, cette méthode ne prend pas en compte la structure globale de l'image et fournit un étiquetage bruité. Pour remédier à cela, un deuxième étage de classification, basé sur des *Conditional Random Field (CRF)* est appliqué. Il fournit une estimation des labels en prenant en compte les contextes spatial et temporel en considérant les distributions de probabilité des classes sur la sphère courante et les sphères voisines dans la séquence d'acquisition au moyen d'une fenêtre glissante. L'implémentation efficace des CRF et de l'algorithme de MAP est issue de [12]. La méthode de classification a été validée sur la séquence d'extérieur comportant 13000 images sphériques haute résolution sur un parcours semi-urbain de 1.6km. Neuf classes ont été définies correspondant aux labels : arbres, ciel, route, bâtiment, panneaux routier, trottoir, marquage au sol, voitures et autre. Un sous ensemble d'images sélectionnées au hasard a été labellisé manuellement pour constituer la base d'apprentissage. La phase d'apprentissage a pris 58 minutes pour l'étape de Random Forest et 43 minutes pour celle de CRF. La figure 5 présente les résultats obtenus sur ce dataset.

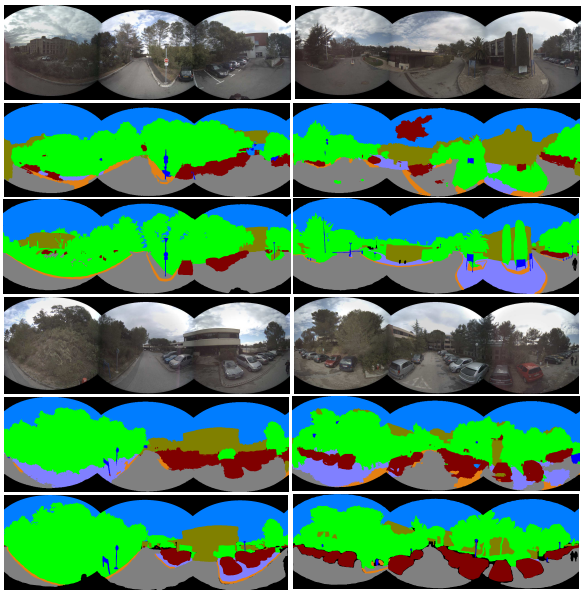


FIGURE 5 – Exemple de labélisation. En haut : panoramique RGB, : Milieu : résultat de la labélisation, En bas : vérité terrain. Les classes sont : arbres (vert), ciel (bleu), route (gris), bâtiment (brun), panneaux routier (bleu foncé), trottoir (violet), marquage au sol (orange), voitures (rouge) et autre (noir).

Finalement, l'information sémantique contenue dans une vue sphérique RGB-DL ("L" pour label) est codée dans le graphe pondéré $\mathcal{G}_o$ des régions labélisées de la sphère RGB-D. Le graphe sémantique $\mathcal{G}_o$ fournit une représentation compacte et robuste du contenu de la scène pour un point de vue donné. Les noeuds sont constitués par les différentes régions sémantiquement consistantes et les arcs représentent la connexité entre régions. A chaque noeud est attaché un attribut caractérisant la région par sa surface en pixels et son excentricité. A chaque arc est associé le nombre de connections entre régions.

2.2 Représentation globale

La représentation globale a pour objet, d'une part de représenter le positionnement des sphères RGB-DL dans l'environnement au niveau métrique et les relations topologiques entre sphères et, d'autre part, de représenter le contenu sémantique par la distribution des occurrences des différents labels dans la scène globale. Le positionnement spatial précis des sphères est obtenu par odométrie visuelle dense complétée par une méthode de fermeture de boucle. Le contenu sémantique est représenté par un modèle probabiliste caractérisant la connaissance a priori existant entre classes.

Positionnement des sphères RGB-DL : Les sphères RGB-DL sont acquises durant un parcours réalisé en conduite manuelle par un opérateur. Au cours de ce parcours les sphères sont enregistrées en continu à 30Hz ce qui correspond à environ une sphère tous les 10cm. Le positionnement 6D précis des sphères RGB-DL est obtenu par une méthode directe de recalage entre les sphères successives de la séquence [15]. Le déplacement entre sphère est défini comme un élément de $\mathbb{SE}(3)$. Le mouvement est paramétré par le torseur de vitesse $\mathbf{x} = \{[\boldsymbol{\omega}, \boldsymbol{\upsilon}] | \upsilon \in \mathbb{R}^3, \hat{\omega} \in so(3)\} \in \mathfrak{se}(3)$: Le vecteur $\mathbf{x}$ est relié à une pose $\mathbf{T}(\mathbf{x})$ par l'application matrice exponentielle.
Le recalage dense sphère à sphère est formalisé par la minimisation d'une fonction de coût portant à la fois sur la photométrie (intensité) et la géométrie (profondeur) :

$$\mathfrak{F}_S = \|e_{\mathcal{I}}\|^2_{D_{\mathcal{I}}} + \lambda^2 \|e_\rho\|^2_{D_{\mathcal{D}}}, \tag{1}$$

où $e_{\mathcal{I}}$ et e_ρ représentent respectivement les erreurs sur l'intensité et sur la profondeur. En développant, on obtient :

$$\begin{aligned} \mathfrak{F}_S = \quad & \frac{1}{2}\sum_i^k \boldsymbol{\eta}_{HUB} \left\| \boldsymbol{\mathcal{I}}\big(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); \boldsymbol{\mathcal{P}}_i^*)\big) - \boldsymbol{\mathcal{I}}^*\big(w(\mathbf{I}; \boldsymbol{\mathcal{P}}_i^*)\big) \right\|^2 \\ + \quad & \frac{\lambda^2}{2}\sum_i^k \boldsymbol{\eta}_{HUB} \left\| n^T\big(\boldsymbol{\mathcal{P}}_i - \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})\boldsymbol{\mathcal{P}}_i^*\big) \right\|^2, \end{aligned} \tag{2}$$

$w(.)$ est la fonction de warping qui projette un point 3D $\boldsymbol{\mathcal{P}}_i$ connaissant sa pose T, sur la sphère courante. $\boldsymbol{\eta}_{HUB}$ est un poids donné par le M-estimateur robuste de Huber [10]. λ, est un paramètre de réglage pondérant les parties photométrique et géométrique de la fonction de coût. n^T est la normale estimée au point $\boldsymbol{\mathcal{P}}_i$ obtenue à partir des points adjacents sur la carte de profondeur.

$\mathbf{x}$ étant commun aux deux parties de l'équation (2), la fonction d'erreur est représentée par le vecteur :

$$\mathbf{e}(\mathbf{x})_S = \begin{bmatrix} \boldsymbol{\eta}_{HUB}\Big(\boldsymbol{\mathcal{I}}\big(w(\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}); \boldsymbol{\mathcal{P}}^*)\big) - \boldsymbol{\mathcal{I}}^*(\boldsymbol{\mathcal{P}}^*)\Big) \\ \boldsymbol{\lambda}\boldsymbol{\eta}_{HUB}\Big(n^T\big(\boldsymbol{\mathcal{P}} - \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x})\boldsymbol{\mathcal{P}}^*\big)\Big) \end{bmatrix} \tag{3}$$

Le Jacobien $\mathbf{J}_S$ de la fonction de coût est donné par :

$$\mathbf{J}_S = \begin{bmatrix} \mathbf{J}_{\mathcal{I}^*}\mathbf{J}_w\mathbf{J}_T \\ \lambda n^T \mathbf{J}_D \end{bmatrix}, \tag{4}$$

où, respectivement, $J_{\mathcal{I}^*}$ est le jacobien par rapport à l'intensité, $\mathbf{J}_w$ est le jacobien par rapport à la fonction de warping, $\mathbf{J}_T$ est le jacobien par rapport à la pose et $\mathbf{J}_D$ est le jacobien par rapport à la profondeur.
De manière identique, les éléments de la fonction de pondération sont regroupés dans la matrice $\mathbf{D}_S$ où $\mathbf{D}_{\boldsymbol{\mathcal{I}}}, \mathbf{D}_{\boldsymbol{\mathcal{D}}} \in \mathbb{R}^{mn \times mn}$ représentent les confiances sur la photométrie et la géométrie pour chaque pixel calculé par les M-estimateurs.

$$\mathbf{D}_S = \begin{bmatrix} \mathbf{D}_{\mathcal{I}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{\mathcal{D}} \end{bmatrix} \quad (5)$$

La minimisation de la fonction de coût est réalisée classiquement par un moindre carré itératif et l'incrément $\mathbf{x}$ est donné par :

$$\mathbf{x} = -(\mathbf{J}_S^T \mathbf{D}_S \mathbf{J}_S)^{-1} \mathbf{J}_S^T \mathbf{D}_S \mathbf{e}(\mathbf{x})_S \quad (6)$$

La pose est mise à jour à chaque itération par la transformation homogène :

$$\widehat{\mathbf{T}} \longleftarrow \widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}), \quad (7)$$

où $\widehat{\mathbf{T}} = [\mathbf{R} \quad \mathbf{t}]$ est la pose estimée courante.
Du fait de la fréquence élevée d'acquisition des sphères RGB-DL, l'information contenue dans deux sphères consécutives est très redondante. Un critère basé sur l'entropie différentielle introduit par Kerl et al. [11] permet de ne conserver dans la représentation que les sphères-clés où l'information est suffisamment différente.

L'estimation de pose étant réalisée par odométrie visuelle, elle est sujette à l'accumulation d'une erreur de dérive au cours du déplacement. Cette erreur est compensée en appliquant la méthode de fermeture de boucle décrite dans [1] que nous ne détaillerons pas ici faute de place. Finalement, la représentation globale hybride métrique/ topologique est constituée d'un graphe où les noeuds sont constitués par les sphères RGB-DL et les arêtes par les poses 6D entre deux sphères. Pour plus de détails sur la construction du graphe de sphères, nous renvoyons le lecteur à [6]. La figure 6 représente la trajectoire estimée par odométrie visuelle et le positionnement des sphères RGB-DL sur la séquence d'extérieur après correction de la dérive par l'algorithme de fermeture de boucle.

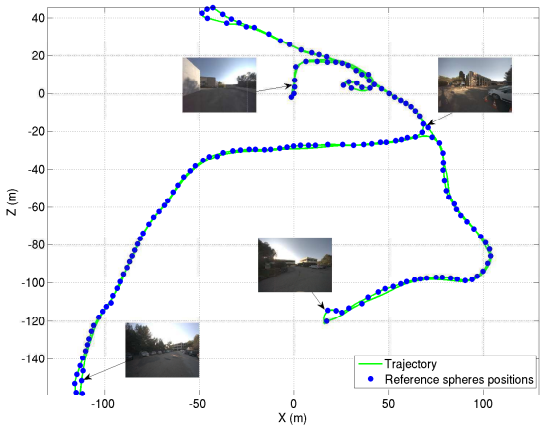


FIGURE 6 – Positionnement des sphères RGB-DL

Modèle sémantique global : La couche conceptuelle (fig.2) est construite à partir des labels contenus dans les sphères RGB-DL. Elle traduit la connaissance sur la scène à l'issue de la labélisation, en termes de relations de voisinage existant entre les différentes classes. Il fournit une probabilité a priori d'avoir un voisinage avec le label l étant donné l'observation d'un pixel avec un label c .

$$P(p_l|p_c) = \frac{\epsilon_{lc}}{\sum_{i \in \mathcal{V}_c} \epsilon_{ic}}$$

où ϵ_{ic} est le poids associé à l'arc entre le noeud de classe i et le noeud de classe c et $\mathcal{V}_c$ les classes connectées avec la classe c.
Ce modèle sémantique permet d'exprimer des relations contextuelles telles que, par exemple, un objet de la classe *voiture* a une forte probabilité d'être en relation avec un objet de la classe *route*.

3 Localisation basée *MTS-Maps*

Par sa structure et son contenu, la représentation MTS-Maps permet de traiter de façon efficace le problème de localisation au niveau global grace à la sémantique, au niveau local en s'appuyant sur la représentation métrique/topologique. Au niveau global, il s'agit de rechercher dans la MTS-Map, la sphère RGB-DL la plus proche du point où est acquis l'image fournie par la caméra que l'on cherche à localiser. Au niveau local, il s'agit d'estimer la pose 6D de la caméra vis à vis de cette sphère. La localisation métrique précise, se formule de façon similaire au problème de positionnement des sphères RGB-DL décrit dans la section 2.2 en appliquant la méthode directe de recalage entre la sphère RGB-DL et l'image fournie par la caméra. Il est alors possible de positionner celle-ci dans le repère global connaissant la position de la sphère RGB-DL dans celui-ci. Cette méthode de localisation a été appliquée à la conduite autonome et validée en environnement urbain dans le cadre du projet ANR CityVIP. Les aspects méthodologiques et les résultats obtenus sont décrits dans [16] auquel nous renvoyons le lecteur.

3.1 Localisation sémantique

Par rapport aux approches éparses basées sur l'utilisation de points d'intérêts associés à des descripteurs locaux, l'utilisation d'une labélisation dense de la sphère RGB-DL permet d'être robuste vis à vis des conditions d'observation pouvant entrainer des occultations (changement de point de vue, présence d'objets dynamiques) ou des changements d'illumination dans l'image. Le problème de localisation est formulé en terme de mise en correspondance de graphes. La méthode de résolution est basée sur l'utilisation d'un arbre d'interprétation qui est une variante de l'approche proposée par Grimson dans [7]. L'arbre d'interprétation est un algorithme efficace qui utilise les

relations entre les noeuds de deux graphes pour accélérer le processus de mise en correspondance. Il utilise deux types de contrantes, unaires et binaires, pour mesurer les similitudes entre graphes.
Soit $\mathcal{G}_1$ et $\mathcal{G}_2$ deux graphes sémantiques (voir section 2.1), une contrainte unaire compare un noeud de $\mathcal{G}_1$ à ceux de $\mathcal{G}_2$. Si la contrainte est vérifiée, ils sont appariés et un score est calculé pour le couple de noeuds. Les paires de noeuds ayant le meilleur score sont ajoutés à la liste $\mathcal{L}$ des meilleurs appariements. Les contraintes binaires sont alors vérifiées sur la liste $\mathcal{L}$. Les contraintes utilisées sont les suivantes :

– *Contraintes unaires :* Elles utilisent les attributs associés aux noeuds : label, excentricité et orientation de l'ellipse englobante à la région dans l'image. La vérification de la contrainte est réalisée au moyen d'un seuil.
– *Contraintes binaires :* Elles utilisent le poids w_i fourni par la matrice d'adjacence de chaque graphe avec :

$$w_i = \sum_{i=1}^{N} p = p_{Ci \to Cj}$$

L'arbre d'interprétation retourne le nombre de noeuds mis en correspondance, la position la plus probable est associée au meilleur score. Pour accélérer le processus, un algorithme d'indexation d'images est utilisé. Les sphères RGB-DL sont organisées dans une structure d'arbre codant l'occurrence des différentes classes (fig.2). Chaque feuille est un sous-ensemble d'images parmi lesquelles le meilleur appariement est trouvé par l'arbre d'interprétation. Cette implémentation permet de réduire de façon drastique le nombre de comparaison entre graphes.

3.2 Résultats

Dans cette section, nous comparons les résultats de notre algorithme à une implémentation récente d'une méthode basée Bag-of-Words (BoW) [1]. Cette méthode construit hors-ligne un dictionnaire sur l'espace des descripteurs. La similarité entre la base de donnée et l'image à classer est évaluée en comptant le nombre de mots visuels communs. Le dictionnaire a été construit avec deux jeux de facteur de branchement et de niveau de profondeur : K=10, L=5 produisant 100000 mots visuels et K=8, L=4 produisant 4096 mots visuels. La stratégie de pondération entre les mots visuels est le *term frequency-inverse document frequency* tf-idf et la norme L1 est utilisée pour le calcul du score (plus de détails sur le choix de ces paramètres dans [4]).

Efficacité temporelle : L'expérimentation consiste à retrouver toutes les images dans la base de données. La table 3.2 présente les temps moyens obtenus avec le BoW, notre algorithme utilisant l'arbre d'interprétation seul (Interp), en utilisant en plus la structure d'arbre d'indexation des sphères (Interp+index) enfin en utilisant uniquement l'indexation (Index). Toutes les implémentations de notre algorithme s'avèrent supérieure à l'implémentation BoW en terme d'efficacité. Cela s'explique, à la fois, par l'utilisation d'une structure d'image qui permet de discriminer rapidement entre bon et mauvais candidats et par la simplicité des tests réalisés (les tests des labels et des attributs de forme sont très rapides). L'utilisation de l'index seul est le plus rapide mais comme il n'encode pas la structure de l'image, il se révèle peu robuste.

Dataset	Temps moyen
BoW K=10, L=5	22ms
BoW K=8, L=4	16ms
Interp	8.40ms
Interp+Index	0.12ms
Index	**54.20**μs

TABLE 1 – Efficacité temporelle des algorithmes

Recherche d'images basée requête de haut niveau : Notre algorithme permet également de traiter des problèmes qui ne peuvent être traités par des approches de type BoW comme, par exemple, la recherche d'image à partir d'une requête de haut niveau. La structure d'arbre utilisé pour l'indexation encode la présence des classes et leur occurrence. De ce fait, il est facile d'exprimer une requête telle que extraire toutes les images où une classe est présente. Il est possible également d'exprimer des requêtes plus complexes mettant en jeu des relations particulières dans l'image ou la scène. La table 3.2 illustre cette capacité et donne les temps de réponse de l'algorithme à ce type de requêtes.

Requête	Temps de réponse
Deux bâtiments	56μs
Pas de voiture	55μs
Des voitures sur la route	0.95ms
Des arbres à la droite des bâtiments	0.86ms

TABLE 2 – Temps de réponse à des requête de haut niveau

Représentation compacte : Enfin, MTS-Maps fournit une représentation complète de l'environnement pouvant être gérée de façon très efficace par un algorithme de navigation. La représentation sémantique sous forme de graphe est très compacte : pour 25 régions sémantiques dans l'image, la mémoire nécessaire pour encoder la matrice d'adjacente est de

1. implémentation présentée dans [4] disponible sur http ://webdiis.unizar.es/ dorian/

1200bytes [2]. Pour chaque noeud, nous ajoutons trois attributs (aire, excentricité et label) ce qui donne un descripteur d'image de 1425bytes (à comparer, par exemple à un descripteur SIFT de 512bytes). Dans l'exemple traité *INRIA dataset* comprenant 13000 images avec une résolution de 2048X665 pixels, la représentation sémantique à une taille de 18.5MBytes (à comparer avec les 52Gbytes des images). Pour ce qui est de la localisation métrique, il est nécessaire de garder en mémoire que le sous ensemble de sphères RGB-DL dans le voisinage de la sphère identifiée par la localisation sémantique et de ce fait, avoir un algorithme de localisation en temps et mémoire constant et indépendant de la taille de la base de donnée.

4 Conclusion

La représentation MTS-Maps est une nouvelle représentation hybride métrique/topologique/sémantique particulièrement bien adaptée pour la navigation dans des environnements à grande échelle. Elle s'appuie sur une représentation, les sphères RGB-DL indépendante du capteur et valide en extérieur et en intérieur. Sa structuration permet de traiter de façon cohérente à la fois le problème de localisation peu précise mais efficace (couche sémantique) nécessaire à la planification d'un déplacement et la localisation temps réel précise indispensable au niveau du contrôle du déplacement (couche métrique).

Références

[1] A. Chapoulie, P. Rives, and D. Filliat. A spherical representation for efficient visual loop closing. In *Proceedings of the 11th workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS 2011)*, Barcelone, Spain, November 2011.

[2] J. Courbon, Y. Mezouar, and P. Martinet. Autonomous navigation of vehicles from a visual memory using a generic camera model. *Intelligent Transport System*, 10 :392–402, 2009.

[3] D. Craciun, N. Paparoditis, and F. Schmitt. Multi-view scans alignment for 3d spherical mosaicing in large-scale unstructured environments. *Computer Vision and Image Understanding*, 114(11) :1248 – 1263, 2010. Special issue on Embedded Vision.

[4] D. Galvez-LoÌĄpez and J.D. Tardos. Bags of binary words for fast place recognition in image sequences. *Robotics, IEEE Transactions on*, 28(5) :1188–1197, Oct 2012.

[5] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1) :3–42, 2006.

[6] T. Gokhool, M. Meilland, P. Rives, and E-F. Moral. A dense map building approach from spherical rgbd images. In *International Conference on Computer Vision Theory and Applications,(VISAPP)*, Lisbon, Portugal, January 2014.

[7] W. Eric L. Grimson. *Object Recognition by Computer : The Role of Geometric Constraints*. MIT Press, Cambridge, MA, USA, 1990.

[8] K. Hammoudi, F. Dornaika, B. Soheilian, and N. Paparoditis. Generating raw polygons of street facades from a 2d urban map and terrestrial laser range data. In *SSSI Australasian Remote Sensing and Photogrammetry Conference*, 2010.

[9] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30 :328–341, 2008.

[10] P.J. Huber. *Robust Statistics*. New York, Wiley, 1981.

[11] C. Kerl, J. Sturm, and D. Cremers. Dense Visual SLAM for RGB-D Cameras. In *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, Tokyo, Japan, 2013.

[12] Philipp Krahenbuhl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In J. Shawe-taylor, R.s. Zemel, P. Bartlett, F.c.n. Pereira, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 109–117. 2011.

[13] F. Lafarge and C. Mallet. Building large urban environments from unstructured point data. In *IEEE International Conference on Computer Vision*, Barcelona, Spain, novembre 2011.

[14] M. Meilland, A.I. Comport, and P. Rives. A spherical robot-centered representation for urban navigation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5196 –5201, oct. 2010.

[15] M. Meilland, A.I. Comport, and P. Rives. Dense visual mapping of large scale environments for real-time localisation. In *IEEE Conference on Intelligent Robots and Systems, IROS'11*, San Fransisco, USA, September 2011.

[16] M. Meilland, P. Rives, and A. I. Comport. Dense rgb-d mapping of large scale environments for real-time localisation and autonomous navigation. In *Intelligent Vehicle (IV'12) Workshop on Navigation, Perception, Accurate Positioning and Mapping for Intelligent Vehicles*, Alcala de Henares, Spain, June 2012.

[17] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. SchrÃűter, L. Murphy, W. Churchill, D. Cole, and I. Reid. Navigating, recognising and describing urban spaces with vision and laser. *The International Journal of Robotics Research*, 2009.

[18] Eric Royer, Maxime Lhuillier, Michel Dhome, and Thierry Chateau. Localization in urban environments : Monocular vision compared to a differential gps sensor. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 114–121, 2005.

[19] N. Tomatis, I. R. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building : a natural integration of topological and metric. *Robotics and Autonomous Systems (RAS*, 44(1) :3–14, 2003.

[20] Wei Zhang and Jana Kosecka. Image based localization in urban environments. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 33–40, 2006.

2. on considère un nombre flottant codé sur 32 bits